

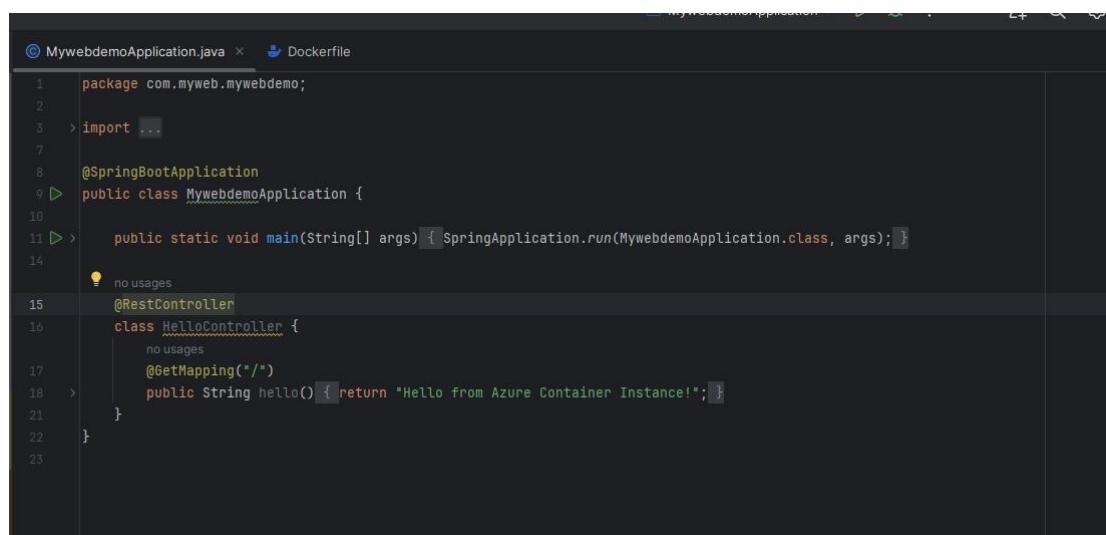
Practical Task 1: Deploy a Docker Container to Azure Container Instances (ACI) via Azure Portal

Requirements:

1. Create a lightweight Docker image for a simple web application (e.g., a Python Flask app) with minimal dependencies to reduce resource usage.
2. Push the Docker image to Azure Container Registry (ACR) using a low-cost storage option.
3. Use a lightweight ACI instance (e.g., B1s) to deploy the Docker container from ACR.
4. Verify the deployment by accessing the web application via the public IP address provided by ACI.
5. Remove the ACI container after verifying the deployment to stop billing.

Actions Taken:

1. Created a Docker image for a Java web application and pushed by docker



The screenshot shows a code editor with two tabs: 'MywebdemoApplication.java' and 'Dockerfile'. The Java file contains a main class 'MywebdemoApplication' with a main method and a controller class 'HelloController' with a single endpoint returning a hello message. The Dockerfile is a standard multi-stage Dockerfile for a Java application.

```
MywebdemoApplication.java
1 package com.myweb.mywebdemo;
2
3 > import ...
4
5 @SpringBootApplication
6 public class MywebdemoApplication {
7
8 >     public static void main(String[] args) { SpringApplication.run(MywebdemoApplication.class, args); }
9
10
11 >     @RestController
12     class HelloController {
13         no usages
14         @GetMapping("/")
15         public String hello() { return "Hello from Azure Container Instance!"; }
16     }
17
18 > }
19
20
21
22
23

Dockerfile
1 FROM maven:3.6.3-slim AS build
2 WORKDIR /app
3 COPY . .
4 RUN mvn clean package
5
6 FROM openjdk:8-jdk-alpine
7 ENV JAVA_OPTS=-Xms512m -Xmx1024m
8 EXPOSE 8080
9 COPY --from=build /app/target/*.jar app.jar
10 ENTRYPOINT ["java", "-Djava.security.egd=file:/dev/./urandom", "-jar", "app.jar"]
```

```
FROM openjdk:17-jdk-slim
WORKDIR /app
COPY target/mywebdemo-0.0.1-SNAPSHOT.jar myapp.jar
EXPOSE 8080
CMD ["java", "-jar", "myapp.jar"]
```

2.Created ACR with name myjavaapp

Microsoft Azure | Search resources, services, and docs (G+)

Home > Microsoft.ContainerRegistry | Overview >

myjavaapp Container registry

Overview

Activity log

Access control (IAM)

Tags

Quick start

Events

Settings

Services

Search

Move

Delete

Essentials

Resource group (move)	myjavaapp.azurecr.io
Location	West Europe
Subscription (move)	Azure subscription 1
Subscription ID	9a6ae428-d8c3-44fe-bdf2-4e08593901a0
Soft delete (Preview)	Disabled
Login server	myjavaapp.azurecr.io
Creation date	1/24/2025, 10:01 PM GMT+2
Provisioning state	Succeeded
Pricing plan	Basic

JSON View

3.Created ACI with name myjavaapp

Microsoft Azure | Search resources, services, and docs (G+)

Home > Microsoft.ContainerInstances-20250124222625 | Overview >

myjavaapp Container instances

Overview

Activity log

Access control (IAM)

Tags

Settings

Monitoring

Automation

Help

Start

Stop

Delete

Refresh

Give feedback

Essentials

Resource group (move)	: Volodymyr-Dibrova
Status	: Running
Location	: West Europe
Subscription (move)	: Azure subscription 1
Subscription ID	: 9a6ae428-d8c3-44fe-bdf2-4e08593901a0
Tags (edit)	: Add tags
SKU	: Standard
OS type	: Linux
IP address (Public)	: 207.6.209.228
FQDN	: helloazurevivi.hug0gfbefakhqzz.westeurope.azurecontainer.io
Container count	: 1

CPU

Memory

JSON View

4.Tested in browser by public IP address



Practical Task 2: Configure Environment Variables in ACI via Azure Portal

Requirements:

1. Modify your Docker image to read configuration values from environment variables, ensuring minimal environmental complexity.
2. Reuse the ACI instance from Task 1 to deploy the container and specify the necessary environment variables.
3. Verify that the application is correctly using the environment variables by checking its output.
4. Remove the ACI container after verifying that the application correctly uses the environment variables.

Actions Taken:

1. Modified the Docker image to work with environment variables and pushed by docker

The image shows a code editor with two files open:

- MywebdemoApplication.java**: A Java class with a main method and a HelloController containing a hello() method that returns the value of the app.message property.
- application.properties**: A properties file defining the application's name, port, and message.

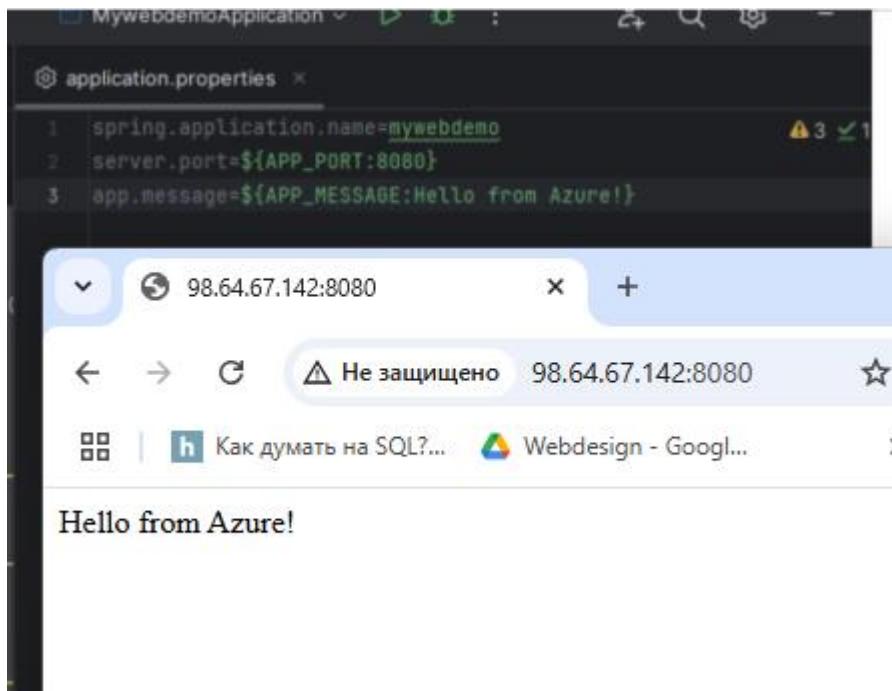
```
MywebdemoApplication.java
import ...
@SpringBootApplication
public class MywebdemoApplication {
    public static void main(String[] args) { SpringApplication.run(
        MywebdemoApplication.class, args);
    }
    @RestController
    class HelloController {
        @Value("${app.message}")
        private String message;
        @GetMapping("/")
        public String hello() { return message; }
    }
}
```

```
application.properties
spring.application.name=mywebdemo
server.port=${APP_PORT:8080}
app.message=${APP_MESSAGE>Hello from Azure!}
```

```

PS C:\Users\User> docker tag mywebdemo myjavaapp.azurecr.io/mywebdemo:v2
PS C:\Users\User> docker push myjavaapp.azurecr.io/mywebdemo:v2
The push refers to repository [myjavaapp.azurecr.io/mywebdemo]
a547e969d43c: Waiting
0798cc789215: Waiting
7b3579d4413b: Waiting
1fe172e4850f: Waiting
44d3aa8d0766: Waiting
6ce99fdf16e8: Waiting
failed to authorize: failed to fetch oauth token: unexpected status from GET request to http://myjavaapp.azurecr.io/oauth2/token?scope=repository%3Amywebdemo%3Apull&scope(repository%3Amywebdemo%3Apull%2Cpush&service=myjavaapp.azurecr.io: 401 Unauthorized
PS C:\Users\User> az acr login --name myjavaapp
Login Succeeded
PS C:\Users\User> docker push myjavaapp.azurecr.io/mywebdemo:v2
The push refers to repository [myjavaapp.azurecr.io/mywebdemo]
44d3aa8d0766: Layer already exists
a547e969d43c: Layer already exists
0798cc789215: Pushed
7b3579d4413b: Pushed
6ce99fdf16e8: Layer already exists
1fe172e4850f: Layer already exists
v2: digest: sha256:43a5b6ac858fb8a242b2348d02c978693b621ee1f416796b3a95b7e3a335e7b6 size: 8
56
PS C:\Users\User> A

```



Practical Task 3: Scale Out with Azure Container Instances via Azure Portal Requirements:

1. Deploy a stateless Docker container to Azure Container Instances using a lightweight configuration (e.g., B1s instances).
2. Manually scale out to the minimum number of instances required (e.g., 2–3) to test load distribution.
3. Stop all ACI instances after completing the testing to reduce ongoing costs.

Actions Taken:

First option: Without ACI

1. Created 3 containers in ACR

```

PS C:\Users\User> for ($x=1; $x -le 3; $x++) {
>>     az container create --resource-group Volodymyr-Dibrova ` 
>>     --name myjavaapp$x ` 
>>     --image myjavaapp.azurecr.io/mywebdemo:v1 ` 
>>     --cpu 1 --memory 1.5 --ports 8080 --os-type Linux ` 
>>     --registry-login-server myjavaapp.azurecr.io ` 
>>     --registry-username myjavaapp ` 
>>     --registry-password "rOxmOlIxoiuhLjq5v08a6Hr4hN90VuZLoi7+R69I6j+ACRA+Emo+"
>> }
{
    "confidentialComputeProperties": null,
    "containerGroupProfile": null,
    "containers": [
        {
            "command": null,
            "configMap": {
                "keyValuePairs": {}
            },
            "environmentVariables": [],
            "image": "myjavaapp.azurecr.io/mywebdemo:v1",
            "instanceView": {
                "currentState": {
                    "detailStatus": "",
                    "exitCode": null
                }
            }
        }
    ]
}

```

2.Checked logs for 3 containers

```
PS C:\Users\User> az container list --resource-group Volodymyr-Dibrova --output table
Name      ResourceGroup    Status   Image          IP:ports     Network  CPU/Memory  OsType  Location
myjavaapp  Volodymyr-Dibrova  Succeeded myjavaapp.azurecr.io/mywebdemo:v1  98.64.53.76:80,8080  Public   1.0 core/1.5 gb Linux  westeurope
myjavaapp1 Volodymyr-Dibrova  Succeeded myjavaapp.azurecr.io/mywebdemo:v1  1.0 core/1.5 gb Linux  westeurope
myjavaapp2 Volodymyr-Dibrova  Succeeded myjavaapp.azurecr.io/mywebdemo:v1  1.0 core/1.5 gb Linux  westeurope
myjavaapp3 Volodymyr-Dibrova  Succeeded myjavaapp.azurecr.io/mywebdemo:v1  1.0 core/1.5 gb Linux  westeurope
PS C:\Users\User> az container logs --resource-group Volodymyr-Dibrova --name myjavaapp1

:: Spring Boot ::      (v3.3.8-SNAPSHOT)

2025-01-25T10:48:27.479Z INFO 19 --- [mywebdemo] [main] c.myweb.mywebdemo.MywebdemoApplication : Starting MywebdemoApplication v0.0.1-SNAPSHOT using Java 17
d by root in /app)
2025-01-25T10:48:31.834Z INFO 19 --- [mywebdemo] [main] c.myweb.mywebdemo.MywebdemoApplication : No active profile set, falling back to 1 default profile: "default"
2025-01-25T10:48:31.834Z INFO 19 --- [mywebdemo] [main] o.s.b.w.embedded.tomcat.TomcatWebServer : Tomcat initialized with port 8080 (http)
2025-01-25T10:48:31.996Z INFO 19 --- [mywebdemo] [main] o.apache.catalina.core.StandardService : Starting service [Tomcat]
2025-01-25T10:48:31.997Z INFO 19 --- [mywebdemo] [main] o.apache.catalina.core.StandardEngine : Starting Servlet engine: [Apache Tomcat/10.1.34]
2025-01-25T10:48:32.427Z INFO 19 --- [mywebdemo] [main] o.a.c.c.C.[Tomcat].[localhost].[/] : Initializing Spring embedded WebApplicationContext
2025-01-25T10:48:32.427Z INFO 19 --- [mywebdemo] [main] o.a.c.c.C.[Tomcat].[localhost].[/] : Initializing Spring Application Context
2025-01-25T10:48:32.427Z INFO 19 --- [mywebdemo] [main] o.s.c.ServletWebServerApplicationContext : Root WebApplicationContext: initialization completed in 3944 ms
2025-01-25T10:49:35.182Z INFO 19 --- [mywebdemo] [main] o.s.b.w.embedded.tomcat.TomcatWebServer : Tomcat started on port 8080 (http) with context path '/'. Started MywebdemoApplication in 7.91 seconds (process running for 10.108)
PS C:\Users\User>
```

```
PS C:\Users\User> az container logs --resource-group Volodymyr-Dibrova --name myjavaapp2

:: Spring Boot ::      (v3.3.8-SNAPSHOT)

2025-01-25T10:49:29.042Z INFO 19 --- [mywebdemo] [main] c.myweb.mywebdemo.MywebdemoApplication : Starting MywebdemoApplication v0.0.1-SNAPSHOT using Java 17.0.2 with PID 19 (/app/myapp.jar starte
d by root in /app)
2025-01-25T10:49:29.658Z INFO 19 --- [mywebdemo] [main] c.myweb.mywebdemo.MywebdemoApplication : No active profile set, falling back to 1 default profile: "default"
2025-01-25T10:49:32.699Z INFO 19 --- [mywebdemo] [main] o.s.b.w.embedded.tomcat.TomcatWebServer : Tomcat initialized with port 8080 (http)
2025-01-25T10:49:32.700Z INFO 19 --- [mywebdemo] [main] o.apache.catalina.core.StandardService : Starting service [Tomcat]
2025-01-25T10:49:32.759Z INFO 19 --- [mywebdemo] [main] o.apache.catalina.core.StandardEngine : Starting Servlet engine: [Apache Tomcat/10.1.34]
2025-01-25T10:49:33.213Z INFO 19 --- [mywebdemo] [main] o.a.c.c.C.[Tomcat].[localhost].[/] : Initializing Spring embedded WebApplicationContext
2025-01-25T10:49:33.221Z INFO 19 --- [mywebdemo] [main] w.s.c.ServletWebServerApplicationContext : Root WebApplicationContext: initialization completed in 3944 ms
2025-01-25T10:49:35.182Z INFO 19 --- [mywebdemo] [main] o.s.b.w.embedded.tomcat.TomcatWebServer : Tomcat started on port 8080 (http) with context path '/'. Started MywebdemoApplication in 7.91 seconds (process running for 10.108)
2025-01-25T10:49:35.231Z INFO 19 --- [mywebdemo] [main] c.myweb.mywebdemo.MywebdemoApplication : Started MywebdemoApplication in 7.91 seconds (process running for 10.108)
PS C:\Users\User> |
```

```
PS C:\Users\User> az container logs --resource-group Volodymyr-Dibrova --name myjavaapp3

:: Spring Boot ::      (v3.3.8-SNAPSHOT)

2025-01-25T10:50:38.149Z INFO 19 --- [mywebdemo] [main] c.myweb.mywebdemo.MywebdemoApplication : Starting MywebdemoApplication v0.0.1-SNAPSHOT using Java 17.0.2 with PID 19 (/app/myapp.jar starte
d by root in /app)
2025-01-25T10:50:38.188Z INFO 19 --- [mywebdemo] [main] c.myweb.mywebdemo.MywebdemoApplication : No active profile set, falling back to 1 default profile: "default"
2025-01-25T10:50:42.567Z INFO 19 --- [mywebdemo] [main] o.s.b.w.embedded.tomcat.TomcatWebServer : Tomcat initialized with port 8080 (http)
2025-01-25T10:50:42.568Z INFO 19 --- [mywebdemo] [main] o.apache.catalina.core.StandardService : Starting service [Tomcat]
2025-01-25T10:50:42.653Z INFO 19 --- [mywebdemo] [main] o.apache.catalina.core.StandardEngine : Starting Servlet engine: [Apache Tomcat/10.1.34]
2025-01-25T10:50:43.121Z INFO 19 --- [mywebdemo] [main] o.a.c.c.C.[Tomcat].[localhost].[/] : Initializing Spring embedded WebApplicationContext
2025-01-25T10:50:43.121Z INFO 19 --- [mywebdemo] [main] w.s.c.ServletWebServerApplicationContext : Root WebApplicationContext: initialization completed in 4787 ms
2025-01-25T10:50:45.297Z INFO 19 --- [mywebdemo] [main] o.s.b.w.embedded.tomcat.TomcatWebServer : Tomcat started on port 8080 (http) with context path '/'
2025-01-25T10:50:45.487Z INFO 19 --- [mywebdemo] [main] c.myweb.mywebdemo.MywebdemoApplication : Started MywebdemoApplication in 9.393 seconds (process running for 12.987)
PS C:\Users\User> |
```

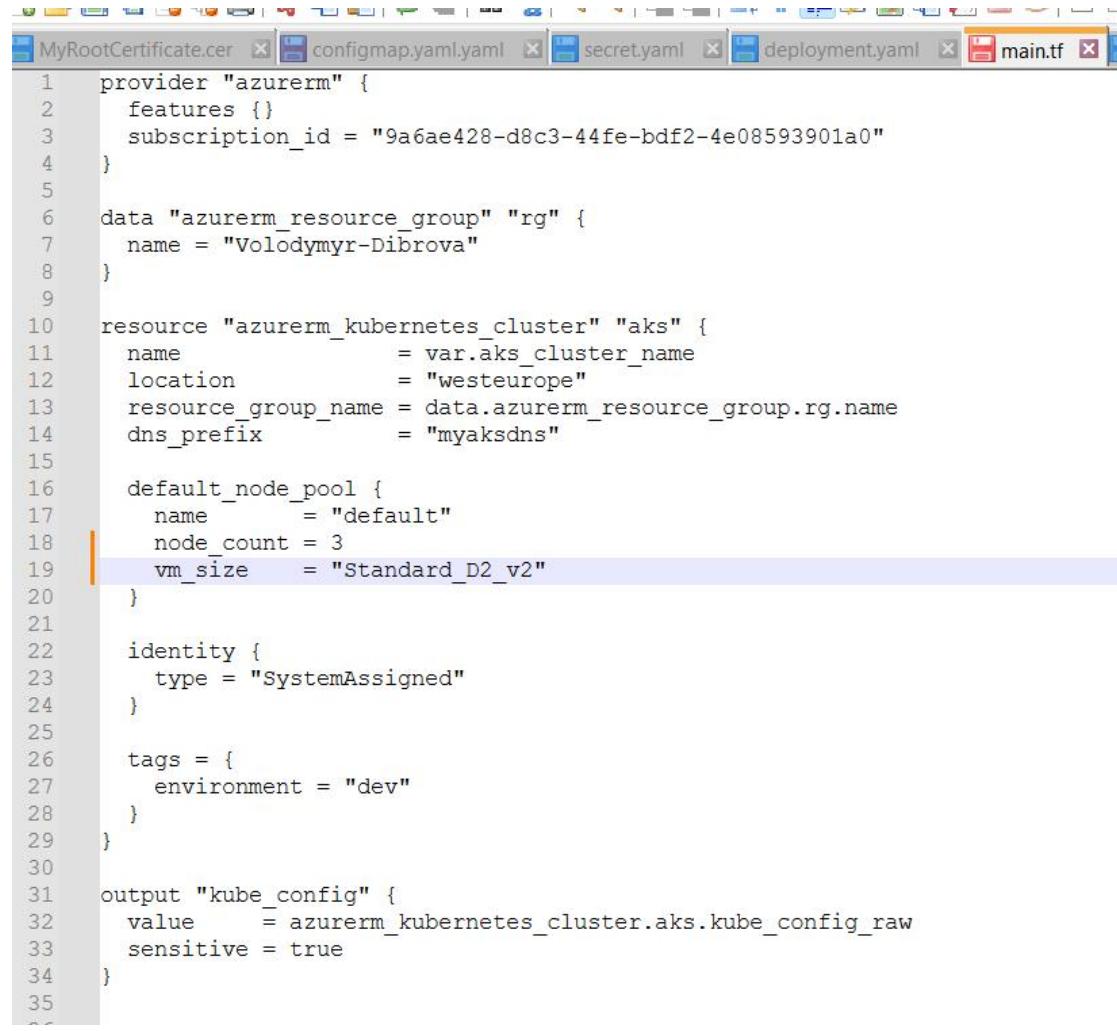
Name	Resource group	Location	Status	OS type	Total c...	Subscription
myjavaapp	Volodymyr-Dibrova	West Europe	Running	Linux	1	Azure subscription 1
myjavaapp1	Volodymyr-Dibrova	West Europe	Running	Linux	1	Azure subscription 1
myjavaapp2	Volodymyr-Dibrova	West Europe	Running	Linux	1	Azure subscription 1
myjavaapp3	Volodymyr-Dibrova	West Europe	Running	Linux	1	Azure subscription 1

3.Tested in browser



Hello from Azure Container Instance!

Second option: with ACI



```

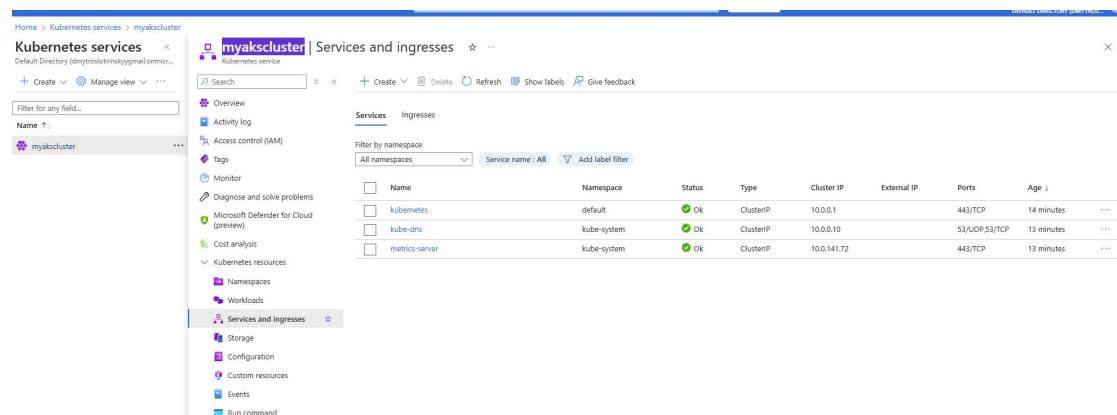
1 provider "azurerm" {
2   features {}
3   subscription_id = "9a6ae428-d8c3-44fe-bdf2-4e08593901a0"
4 }
5
6 data "azurerm_resource_group" "rg" {
7   name = "Volodymyr-Dibrova"
8 }
9
10 resource "azurerm_kubernetes_cluster" "aks" {
11   name           = var.aks_cluster_name
12   location       = "westeurope"
13   resource_group_name = data_azurerm_resource_group.rg.name
14   dns_prefix     = "myaksdns"
15
16   default_node_pool {
17     name      = "default"
18     node_count = 3
19     vm_size    = "Standard_D2_v2"
20   }
21
22   identity {
23     type = "SystemAssigned"
24   }
25
26   tags = {
27     environment = "dev"
28   }
29 }
30
31 output "kube_config" {
32   value    = azurerm_kubernetes_cluster.aks.kube_config_raw
33   sensitive = true
34 }
35
36

```

```

PS E:\projects\terraformProj\terraformProj> az aks get-credentials --resource-group Volodymyr-Dibrova --name myakscluster --overwrite-existing
Merged "myakscluster" as current context in C:\Users\User\.kube\config
PS E:\projects\terraformProj> kubectl get nodes
NAME           STATUS   ROLES   AGE    VERSION
aks-default-88566151-vmss000000  Ready    <none>   9m37s   v1.38.7
aks-default-88566151-vmss000001  Ready    <none>   9m40s   v1.38.7
aks-default-88566151-vmss000002  Ready    <none>   9m37s   v1.38.7
PS E:\projects\terraformProj\terraformProj>

```



Name	Namespace	Status	Type	Cluster IP	External IP	Ports	Age
kubernetes	default	Ok	ClusterIP	10.0.0.1		443/TCP	14 minutes
kube-dns	kube-system	Ok	ClusterIP	10.0.0.10		53/UDP,53/TCP	13 minutes
metrics-server	kube-system	Ok	ClusterIP	10.0.141.72		443/TCP	13 minutes

```

PS E:\projects\terraformProj> az aks get-credentials --resource-group Volodymyr-Dibrova --name myakscluster --overwrite-existing
Merged "myakscluster" as current context in C:\Users\User\.kube\config
PS E:\projects\terraformProj> kubectl get nodes
NAME STATUS ROLES AGE VERSION
aks-default-88566151~vmss000000 Ready <none> 9m37s v1.30.7
aks-default-88566151~vmss000001 Ready <none> 9m40s v1.30.7
aks-default-88566151~vmss000002 Ready <none> 9m37s v1.30.7
PS E:\projects\terraformProj> az acr credential show --name myjavaapp
{
  "passwords": [
    {
      "name": "password",
      "value": "rOxm0lXoiuhLjq5v08a6Hr4hN90VuZLoi7+R69I6j+ACRA+Emo+"
    },
    {
      "name": "password2",
      "value": "aJLardyzCEPBBKwdH197wOx0bfu/jVTEqc3xMmSBta+ACRBZwPKB"
    }
  ],
  "username": "myjavaapp"
}
PS E:\projects\terraformProj> kubectl create secret docker-registry acr-secret \
--docker-server=myjavaapp.azurecr.io \
--docker-username=myjavaapp \
--docker-password="rOxm0lXoiuhLjq5v08a6Hr4hN90VuZLoi7+R69I6j+ACRA+Emo+"
secret/acr-secret created
PS E:\projects\terraformProj> |
PS E:\projects\terraformProj\terraform> kubectl get pods
NAME READY STATUS RESTARTS AGE
mywebdemo-deployment-dc9b99db4-5t2dv 0/1 ContainerCreating 0 5s
mywebdemo-deployment-dc9b99db4-cls9d 0/1 ContainerCreating 0 5s
mywebdemo-deployment-dc9b99db4-qhcfz 0/1 ContainerCreating 0 5s
PS E:\projects\terraformProj\terraform> kubectl get deployments
NAME READY UP-TO-DATE AVAILABLE AGE
mywebdemo-deployment 3/3 3 3 13s
PS E:\projects\terraformProj\terraform> |

```

Logs for every containers:

```

PS E:\projects\terraformProj\nodeWebApp> kubectl logs mywebdemo-deployment-dc9b99db4-5t2dv
```
:: Spring Boot :: (v3.3.8-SNAPSHOT)

2025-01-29T14:50:39.442Z INFO 1 --- [mywebdemo] [main] c.myweb.mywebdemo.MywebdemoApplication : Starting MywebdemoApplication v0.0.1-SNAPSHOT
using Java 17.0.2 with PID 1 (/app/myapp.jar started by root in /app)
2025-01-29T14:50:39.455Z INFO 1 --- [mywebdemo] [main] c.myweb.mywebdemo.MywebdemoApplication : No active profile set, falling back to 1 defau
lt profile: "default"
2025-01-29T14:50:40.792Z INFO 1 --- [mywebdemo] [main] o.s.b.w.embedded.tomcat.TomcatWebServer : Tomcat initialized with port 8080 (http)
2025-01-29T14:50:40.809Z INFO 1 --- [mywebdemo] [main] o.apache.catalina.core.StandardService : Starting service [Tomcat]
2025-01-29T14:50:40.810Z INFO 1 --- [mywebdemo] [main] o.apache.catalina.core.StandardEngine : Starting Servlet engine: [Apache Tomcat/10.1.3
4]
2025-01-29T14:50:41.021Z INFO 1 --- [mywebdemo] [main] o.a.c.c.C.[Tomcat].[localhost].[] : Initializing Spring embedded WebApplicationCon
text
2025-01-29T14:50:41.025Z INFO 1 --- [mywebdemo] [main] w.s.c.ServletWebServerApplicationContext : Root WebApplicationContext: initialization com
pleted in 1476 ms
2025-01-29T14:50:42.135Z INFO 1 --- [mywebdemo] [main] o.s.b.w.embedded.tomcat.TomcatWebServer : Tomcat started on port 8080 (http) with contex
t path '/'
2025-01-29T14:50:42.163Z INFO 1 --- [mywebdemo] [main] c.myweb.mywebdemo.MywebdemoApplication : Started MywebdemoApplication in 3.409 seconds
(process running for 4.178)
PS E:\projects\terraformProj\nodeWebApp> kubectl logs mywebdemo-deployment-dc9b99db4-cls9d
```
:: Spring Boot ::   (v3.3.8-SNAPSHOT)

2025-01-29T14:50:39.366Z INFO 1 --- [mywebdemo] [main] c.myweb.mywebdemo.MywebdemoApplication : Starting MywebdemoApplication v0.0.1-SNAPSHOT
using Java 17.0.2 with PID 1 (/app/myapp.jar started by root in /app)
2025-01-29T14:50:39.370Z INFO 1 --- [mywebdemo] [main] c.myweb.mywebdemo.MywebdemoApplication : No active profile set, falling back to 1 defau
lt profile: "default"
2025-01-29T14:50:40.849Z INFO 1 --- [mywebdemo] [main] o.s.b.w.embedded.tomcat.TomcatWebServer : Tomcat initialized with port 8080 (http)
2025-01-29T14:50:40.869Z INFO 1 --- [mywebdemo] [main] o.apache.catalina.core.StandardService : Starting service [Tomcat]
2025-01-29T14:50:40.869Z INFO 1 --- [mywebdemo] [main] o.apache.catalina.core.StandardEngine : Starting Servlet engine: [Apache Tomcat/10.1.3
4]
2025-01-29T14:50:41.043Z INFO 1 --- [mywebdemo] [main] o.a.c.c.C.[Tomcat].[localhost].[] : Initializing Spring embedded WebApplicationCon
text
2025-01-29T14:50:41.045Z INFO 1 --- [mywebdemo] [main] w.s.c.ServletWebServerApplicationContext : Root WebApplicationContext: initialization com
pleted in 1604 ms
2025-01-29T14:50:42.072Z INFO 1 --- [mywebdemo] [main] o.s.b.w.embedded.tomcat.TomcatWebServer : Tomcat started on port 8080 (http) with contex
t path '/'
2025-01-29T14:50:42.098Z INFO 1 --- [mywebdemo] [main] c.myweb.mywebdemo.MywebdemoApplication : Started MywebdemoApplication in 3.256 seconds
(process running for 4.065)
2025-01-29T14:58:47.408Z INFO 1 --- [mywebdemo] [io-8080-exec-10] o.a.c.c.C.[Tomcat].[localhost].[] : Initializing Spring DispatcherServlet 'dispat
cherServlet'
2025-01-29T14:58:47.409Z INFO 1 --- [mywebdemo] [io-8080-exec-10] o.s.web.servlet.DispatcherServlet : Initializing Servlet 'dispatcherServlet'
2025-01-29T14:58:47.410Z INFO 1 --- [mywebdemo] [io-8080-exec-10] o.s.web.servlet.DispatcherServlet : Completed initialization in 1 ms
PS E:\projects\terraformProj\nodeWebApp> kubectl logs mywebdemo-deployment-dc9b99db4-qhcfz
```

```

```

PS E:\projects\terraformProj\nodeWebApp> kubectl logs mywebdemo-deployment-dc9b99db4-cls9d
```
:: Spring Boot ::   (v3.3.8-SNAPSHOT)

2025-01-29T14:50:39.366Z INFO 1 --- [mywebdemo] [main] c.myweb.mywebdemo.MywebdemoApplication : Starting MywebdemoApplication v0.0.1-SNAPSHOT
using Java 17.0.2 with PID 1 (/app/myapp.jar started by root in /app)
2025-01-29T14:50:39.370Z INFO 1 --- [mywebdemo] [main] c.myweb.mywebdemo.MywebdemoApplication : No active profile set, falling back to 1 defau
lt profile: "default"
2025-01-29T14:50:40.849Z INFO 1 --- [mywebdemo] [main] o.s.b.w.embedded.tomcat.TomcatWebServer : Tomcat initialized with port 8080 (http)
2025-01-29T14:50:40.869Z INFO 1 --- [mywebdemo] [main] o.apache.catalina.core.StandardService : Starting service [Tomcat]
2025-01-29T14:50:40.869Z INFO 1 --- [mywebdemo] [main] o.apache.catalina.core.StandardEngine : Starting Servlet engine: [Apache Tomcat/10.1.3
4]
2025-01-29T14:50:41.043Z INFO 1 --- [mywebdemo] [main] o.a.c.c.C.[Tomcat].[localhost].[] : Initializing Spring embedded WebApplicationCon
text
2025-01-29T14:50:41.045Z INFO 1 --- [mywebdemo] [main] w.s.c.ServletWebServerApplicationContext : Root WebApplicationContext: initialization com
pleted in 1604 ms
2025-01-29T14:50:42.072Z INFO 1 --- [mywebdemo] [main] o.s.b.w.embedded.tomcat.TomcatWebServer : Tomcat started on port 8080 (http) with contex
t path '/'
2025-01-29T14:50:42.098Z INFO 1 --- [mywebdemo] [main] c.myweb.mywebdemo.MywebdemoApplication : Started MywebdemoApplication in 3.256 seconds
(process running for 4.065)
2025-01-29T14:58:47.408Z INFO 1 --- [mywebdemo] [io-8080-exec-10] o.a.c.c.C.[Tomcat].[localhost].[] : Initializing Spring DispatcherServlet 'dispat
cherServlet'
2025-01-29T14:58:47.409Z INFO 1 --- [mywebdemo] [io-8080-exec-10] o.s.web.servlet.DispatcherServlet : Initializing Servlet 'dispatcherServlet'
2025-01-29T14:58:47.410Z INFO 1 --- [mywebdemo] [io-8080-exec-10] o.s.web.servlet.DispatcherServlet : Completed initialization in 1 ms
PS E:\projects\terraformProj\nodeWebApp> kubectl logs mywebdemo-deployment-dc9b99db4-qhcfz
```

```

```

PS E:\projects\terraformProj\nodeWebApp> kubectl logs mywebdemo-deployment-dc9b99db4-qhcfcz

:: Spring Boot :: (v3.3.8-SNAPSHOT)

2025-01-29T14:50:39.614Z INFO 1 --- [mywebdemo] [main] c.myweb.mywebdemo.MywebdemoApplication : Starting MywebdemoApplication v0.0.1-SNAPSHOT
using Java 17.0.2 with PID 1 (/app/myapp.jar started by root in /app)
2025-01-29T14:50:39.617Z INFO 1 --- [mywebdemo] [main] c.myweb.mywebdemo.MywebdemoApplication : No active profile set, falling back to 1 defau
lt profile: "default"
2025-01-29T14:50:40.945Z INFO 1 --- [mywebdemo] [main] o.s.b.w.embedded.tomcat.TomcatWebServer : Tomcat initialized with port 8080 (http)
2025-01-29T14:50:40.962Z INFO 1 --- [mywebdemo] [main] o.apache.catalina.core.StandardService : Starting service [Tomcat]
2025-01-29T14:50:40.963Z INFO 1 --- [mywebdemo] [main] o.apache.catalina.core.StandardEngine : Starting Servlet engine: [Apache Tomcat/10.1.3
4]
2025-01-29T14:50:41.242Z INFO 1 --- [mywebdemo] [main] o.a.c.c.C.[Tomcat].[localhost].[] : Initializing Spring embedded WebApplicationCon
text
2025-01-29T14:50:41.244Z INFO 1 --- [mywebdemo] [main] w.s.c.ServletWebServerApplicationContext : Root WebApplicationContext: initialization com
pleted in 1568 ms
2025-01-29T14:50:42.174Z INFO 1 --- [mywebdemo] [main] o.s.b.w.embedded.tomcat.TomcatWebServer : Tomcat started on port 8080 (http) with contex
t path '/'
2025-01-29T14:50:42.194Z INFO 0 --- [mywebdemo] [main] c.myweb.mywebdemo.MywebdemoApplication : Started MywebdemoApplication in 3.195 seconds
(processes running in 3.922)
2025-01-29T14:50:42.194Z INFO 0 --- [mywebdemo] [nio-8080-exec-4] o.a.c.c.C.[Tomcat].[localhost].[] : Initializing Spring DispatcherServlet 'dispatc
herServlet'
2025-01-29T14:50:58.660Z INFO 0 --- [mywebdemo] [nio-8080-exec-4] o.s.web.servlet.DispatcherServlet : Initializing Servlet 'dispatcherServlet'
2025-01-29T14:50:58.660Z INFO 0 --- [mywebdemo] [nio-8080-exec-4] o.s.web.servlet.DispatcherServlet : Completed initialization in 1 ms
PS E:\projects\terraformProj\nodeWebApp>

```

### Check load balancing between 3 pods

```

PS E:\projects\terraformProj\nodeWebApp> while ($true) { curl http://40.74.47.105; Start-Sleep -Seconds 1 }

Content : Hello from Azure!
StatusDescription :
StatusCode : 200
RawContent : HTTP/1.1 200
Keep-Alive: timeout=60
Connection: keep-alive
Content-Length: 17
Content-Type: text/plain;charset=UTF-8
Date: Wed, 29 Jan 2025 15:08:06 GMT
Forms : {}
Headers : {[Keep-Alive, timeout=60], [Connection, keep-alive], [Content-Length, 17], [Content-Type, text/plain;charset=UTF-8]}
Images : {}
InputFields : {}
Links : {}
Object :
ParsedHtml : System.__ComObject
RawContentLength : 17
StatusDescription : Hello from Azure!
Content : Hello from Azure!
StatusCode : 200
RawContent : HTTP/1.1 200
Keep-Alive: timeout=60
Connection: keep-alive
Content-Length: 17
Content-Type: text/plain;charset=UTF-8
Date: Wed, 29 Jan 2025 15:08:07 GMT
Forms : {}
Headers : {[Content-Length, 17], [Content-Type, text/plain;charset=UTF-8], [Date, Wed, 29 Jan 2025 15:08:07 GMT]}
Images : {}
InputFields : {}
Links : {}
Object :
ParsedHtml : System.__ComObject
RawContentLength : 17
StatusDescription : Hello from Azure!
Content : Hello from Azure!
StatusCode : 200
RawContent : HTTP/1.1 200
Keep-Alive: timeout=60
Connection: keep-alive
Content-Length: 17
Content-Type: text/plain;charset=UTF-8
Date: Wed, 29 Jan 2025 15:08:09 GMT
Forms : {}
Headers : {}
Images : {}
InputFields : {}
Links : {}
Object :

```

### Practical Task 4: Secure a Docker Container in ACI with Managed Identity via Azure Portal

#### Requirements:

1. Deploy a Docker container to Azure Container Instances using the existing lightweight ACI setup from previous tasks.
2. Configure a Managed Identity for the ACI and securely access an Azure service (e.g., Azure Key Vault) with minimal permissions and access scope.
3. Retrieve only a single secret from Azure Key Vault for testing purposes.
4. Remove the ACI container after verifying secure access.

#### Actions Taken:

## 1.Created app for getting Secret from Azure

```

public class SecretController {
 private final SecretClient secretClient;

 public SecretController(SecretClient secretClient) {
 this.secretClient = secretClient;
 }

 @GetMapping("/secret")
 public String getSecret() {
 try {
 return secretClient.getSecret("MySecret").getValue();
 } catch (Exception e) {
 return "Failed to retrieve secret.";
 }
 }
}

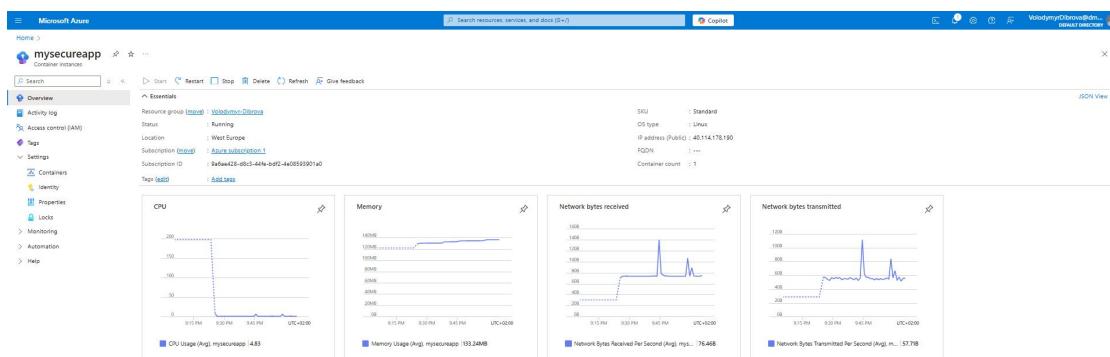
```

```

spring.application.name=mywebdemo
server.port=${APP_PORT:8080}
app.message=${APP_MESSAGE>Hello from Azure!}

azure.keyvault.uri=https://mykeyvault2025.vault.azure.net
azure.keyvault.secret=MySecret

```



## 2.Created Key Vault and added Role Assignment (Key Vault Administrator)

Key vault: myKeyvault2025

Overview

Activity log

Access control (IAM)

Tags

Diagnose and solve problems

Access policies

Events

Objects

Keys

Secrets

Certificates

Properties

Monitoring

Automation

Help

Get started Properties Monitoring Tools + SDKs Tutorials

Manage keys and secrets used by apps and services

Control access to key vault

Enable logging and set up alerts

Turn on recovery options

Access control (IAM)

| Action          | Principal                        | Role              | Subscription                 | Resource                 |
|-----------------|----------------------------------|-------------------|------------------------------|--------------------------|
| Add             | Dmytro Slobotivskyi              | User              | Owner                        | Subscription (Inherited) |
|                 | Foreign Principal for FOREIGN CO | Foreign principal | Owner                        | Subscription (Inherited) |
| Contributor (1) | Capgemini Azure Course 2025      | Group             | Contributor                  | Subscription (Inherited) |
|                 | Advisor Reviews Reader (1)       | App               | Advisor Reviews Reader       | Subscription (Inherited) |
|                 | besaha-principal1                | App               | Compute Recommendations Role | Subscription (Inherited) |
|                 | besaha-principal                 | App               | Compute Recommendations Role | Subscription (Inherited) |
|                 | Key Vault Administrator (1)      | User              | Key Vault Administrator      | This resource            |
|                 | Volodymyr Dibrova                | User              |                              |                          |

**Properties**

- Created: 1/26/2021, 9:33:08 PM
- Updated: 1/26/2021, 9:33:08 PM
- Secret identifier: <https://mykeyvault2021.vault.azure.net/secrets/MySecret/70bbaa7b46ad431fbda96f32bfb1bc7>
- Settings:
  - Set activation date:
  - Set expiration date:
- Enabled: Yes
- Tags: 0 tags
- Secret:
  - Content type (optional):
  - Hide Secret Value**
- Secret value: SuperSecretValue

**Tags**: 0 tags

**Secret**

**Content type (optional)**

**Hide Secret Value**

**Secret value**: SuperSecretValue

**mywebdemo**

Repository: mywebdemo

Last updated date: 1/26/2021, 10:03 PM GMT+2

Tag count: 3  
Manifest count: 0

| Tags | Digest                                                                 | Last modified              |
|------|------------------------------------------------------------------------|----------------------------|
| v3   | sha256:98320ae9fc1409020995c1a3aa1ee16ec048501a95041a534c83936974d8d12 | 1/26/2021, 10:03 PM, GMT+2 |
| v2   | sha256:431b0de8389fb8a24262348b02d976693d21ee141f79563a9507e335476     | 1/26/2021, 12:18 PM, GMT+2 |
| v1   | sha256:05f70cc1e544e30a4986a73328a3984b909a340a7e97a77d11a11121ca      | 1/24/2021, 10:13 PM, GMT+2 |

Tested in browser by v3 - retrieved secret

108.141.3.117:8080/secret

SuperSecretValue

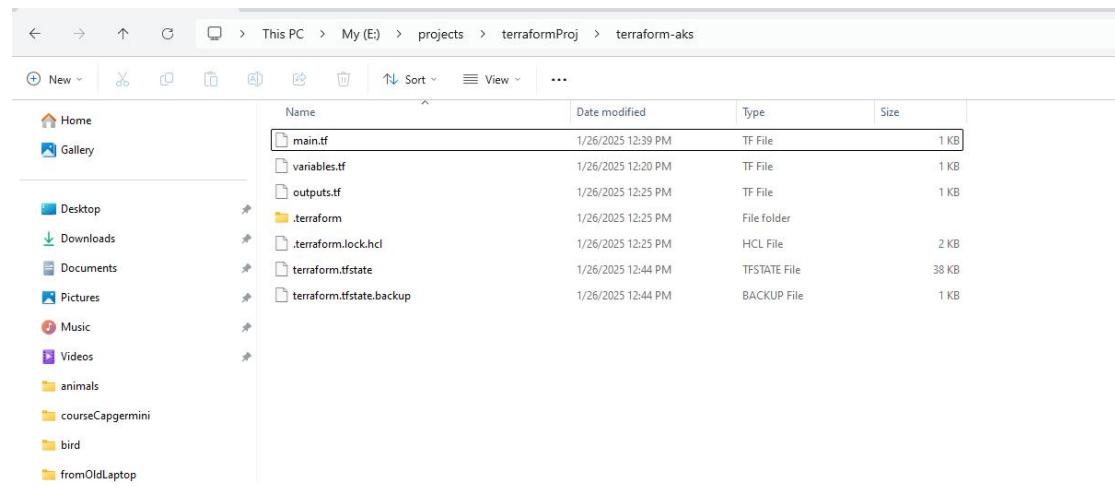
## Practical Task 5: Deploy a Kubernetes Cluster with AKS via Azure Portal Requirements:

1. Create an Azure Kubernetes Service (AKS) cluster with the smallest VM size (e.g., B2s) and the minimum number of nodes (e.g., 1–2).

2. Connect to the AKS cluster using Azure Cloud Shell with kubectl.
3. Deploy a lightweight Nginx application for verification.
4. Delete the AKS cluster immediately after testing to avoid additional VM and cluster

**Actions Taken:**

1.created cluster using Terraform



The screenshot shows a code editor interface with a dark theme. The top menu bar includes File, Edit, Selection, View, Go, Run, Terminal, and Help. Below the menu is a tab bar with 'Welcome' (highlighted in blue), 'main.tf' (highlighted in purple), 'variables.tf' (grey), and 'outputs.tf' (grey). The main editor area displays the following Terraform code:

```
E: > projects > terraformProj > terraform-aks > main.tf
1 provider "azurerm" {
2 features {}
3 subscription_id = "9a6ae428-d8c3-44fe-bdf2-4e08593901a0"
4 }
5
6 data "azurerm_resource_group" "rg" {
7 name = "Volodymyr-Dibrova"
8 }
9
10 resource "azurerm_kubernetes_cluster" "aks" [
11 name = var.aks_cluster_name
12 location = "westeurope"
13 resource_group_name = "Volodymyr-Dibrova"
14 dns_prefix = "myaksdns"
15]
16
17 default_node_pool {
18 name = "default"
19 node_count = var.node_count
20 vm_size = var.vm_size
21 }
22
23 identity {
24 type = "SystemAssigned"
25 }
26
27 tags = {
28 environment = "dev"
29 }
30]
31
32
33 output "kube_config" {
34 value = azurerm_kubernetes_cluster.aks.kube_config_raw
35 sensitive = true
36 }
37
```

File Edit Selection View Go Run Terminal Help

Welcome main.tf variables.tf ● outputs.tf

E: > projects > terraformProj > terraform-aks > variables.tf

```
1
2 variable "aks_cluster_name" {
3 default = "myakscluster"
4 }
5
6 variable "location" {
7 default = "West Europe"
8 }
9
10 variable "node_count" {
11 default = 1
12 }
13
14 variable "vm_size" {
15 default = "Standard_B2s"
16 }
17
```

File Edit Selection View Go Run Terminal Help

Welcome main.tf variables.tf ● outputs.tf X

E: > projects > terraformProj > terraform-aks > outputs.tf

```
1
2 output "aks_fqdn" {
3 value = azurerm_kubernetes_cluster.aks.fqdn
4 }
5
```

Microsoft Azure

Kubernetes services myakscluster

Overview

Resource group: myakscluster

Power state: Running

Cluster operation status: Succeeded

Subscription: Azure Subscription 1

Location: West Europe

Subscription ID: 9ed4421-dc13-449c-bd04-e0859901a0

Tags (1): environment: dev

Kubernetes version: 1.30.7

API server address: myakscluster-hub1.ingress.westeurope.azurefdn.io

Network configuration: Azure CN1 Overlay

Node pools: 1 node pool

Container registries: Attach a registry

Networking

API server address: myakscluster-hub1.ingress.westeurope.azurefdn.io

Azure CN1 Overlay

Pod CIDR: 10.244.0.0/16

Service CIDR: 10.0.0.0/16

DNIS service IP: 10.0.0.1

Client database: Not enabled

Network Policy: None

Successfully connected

```
PS /home/volodymyr> az aks get-credentials --resource-group Volodymyr-Dibrova --name myakscluster
Merged "myakscluster" as current context in /home/volodymyr/.kube/config
PS /home/volodymyr> kubectl get nodes
NAME STATUS ROLES AGE VERSION
aks-default-26255206-vmss000000 Ready <none> 13m v1.30.7
PS /home/volodymyr>
```

The screenshot shows a code editor interface with several tabs at the top: Welcome, main.tf, variables.tf, nginx-deployment.yaml (which is the active tab), and nginx-service.yaml. The code in the editor is a Kubernetes Deployment manifest:

```
1 apiVersion: apps/v1
2 kind: Deployment
3 metadata:
4 name: nginx-deployment
5 labels:
6 app: nginx
7 spec:
8 replicas: 2
9 selector:
10 matchLabels:
11 app: nginx
12 template:
13 metadata:
14 labels:
15 app: nginx
16 spec:
17 containers:
18 - name: nginx
19 image: nginx:latest
20 ports:
21 - containerPort: 80
```

The screenshot shows a code editor interface with several tabs at the top: Welcome, main.tf, variables.tf, nginx-deployment.yaml, and nginx-service.yaml (which is the active tab). The code in the editor is a Kubernetes Service manifest:

```
1 apiVersion: v1
2 kind: Service
3 metadata:
4 name: nginx-service
5 spec:
6 selector:
7 app: nginx
8 ports:
9 - protocol: TCP
10 port: 80
11 targetPort: 80
12 type: LoadBalancer
```

```

PS E:\projects\terraformProj\terraform-aks> kubectl apply -f nginx-deployment.yaml
deployment.apps/nginx-deployment created
PS E:\projects\terraformProj\terraform-aks> kubectl apply -f nginx-service.yaml
service/nginx-service created
PS E:\projects\terraformProj\terraform-aks> kubectl get deployments
NAME READY UP-TO-DATE AVAILABLE AGE
nginx-deployment 2/2 2 2 41s
PS E:\projects\terraformProj\terraform-aks> kubectl get pods
NAME READY STATUS RESTARTS AGE
nginx-deployment-576c6b7b6-6pkt6 1/1 Running 0 53s
nginx-deployment-576c6b7b6-zbqq4 1/1 Running 0 53s
PS E:\projects\terraformProj\terraform-aks> kubectl get services
NAME TYPE CLUSTER-IP EXTERNAL-IP PORT(S) AGE
kubernetes ClusterIP 10.0.0.1 <none> 443/TCP 24m
nginx-service LoadBalancer 10.0.184.157 20.8.228.57 80:31934/TCP 33s
PS E:\projects\terraformProj\terraform-aks> kubectl get svc nginx-service
NAME TYPE CLUSTER-IP EXTERNAL-IP PORT(S) AGE
nginx-service LoadBalancer 10.0.184.157 20.8.228.57 80:31934/TCP 44s
PS E:\projects\terraformProj\terraform-aks>

```

### Nginx Deployment:

Name: nginx-deployment

Status: Both pods (2/2) are **running correctly**

Age: 41 seconds.

### Pods (containers):

Both pods are running (Running)

Age: 53 seconds.

### Nginx Service:

Type: LoadBalancer

Internal IP (CLUSTER-IP): 10.0.184.157

External IP (EXTERNAL-IP): **20.8.228.57**

Port: 80:31934/TCP

Age: 44 seconds.

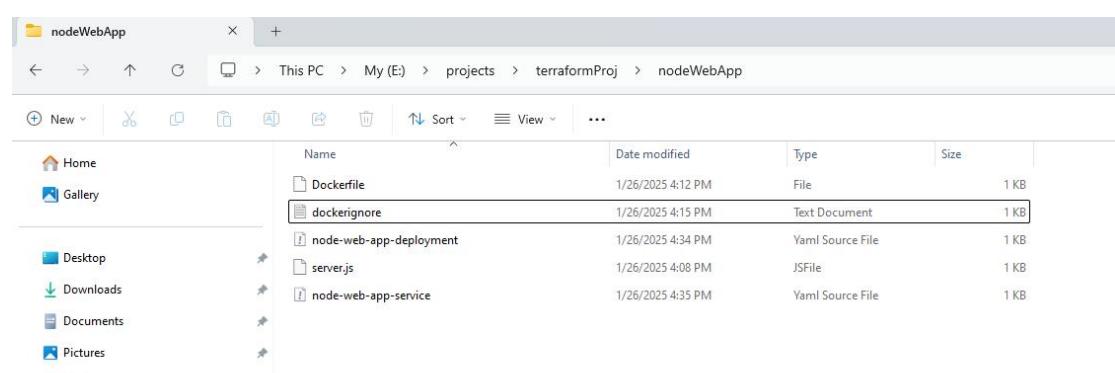


## Practical Task 6: Deploy a Containerized Application on AKS

### Requirements:

1. Build a lightweight Docker image for a simple web application (e.g., a Node.js app with minimal dependencies) and push it to Azure Container Registry (ACR).
2. Reuse the AKS cluster from Task 5 to deploy the application using a Kubernetes deployment and service manifest file.
3. Test the application for a limited time and remove the deployment afterward.

### Actions Taken:



```
PS E:\projects\terraformProj\nodeWebApp> docker images
REPOSITORY TAG IMAGE ID CREATED SIZE
myjavaapp.azurecr.io/node-web-app v1 43242ddf49e0 2 minutes ago 181MB
<none> <none> dfa7d3ec182d 18 hours ago 722MB
mywebdemo latest dd8b1164f06f 18 hours ago 722MB
myjavaapp.azurecr.io/mywebdemo v3 dd8b1164f06f 18 hours ago 722MB
myjavaapp.azurecr.io/mywebdemo v2 9f82baefe0c1 18 hours ago 722MB
<none> <none> 43a5b6ac858f 28 hours ago 680MB
myapp latest b5f709c1e5e4 43 hours ago 680MB
myjavaapp.azurecr.io/mywebdemo v1 b5f709c1e5e4 43 hours ago 680MB
workerapi-gateway latest 59b2d00c1d86 5 weeks ago 681MB
worker latest 2d050d945d91 5 weeks ago 681MB
frontend latest d7bee6c4b85e 5 weeks ago 1.36GB
gcr.io/k8s-minikube/kicbase v0.0.45 e7c9bc3bc515 4 months ago 1.81GB
gcr.io/k8s-minikube/kicbase <none> 81df28859520 4 months ago 1.81GB
PS E:\projects\terraformProj\nodeWebApp> docker push myjavaapp.azurecr.io/node-web-app:v1
The push refers to repository [myjavaapp.azurecr.io/node-web-app]
6504e29600c8: Waiting
c1875295cf1f: Waiting
5650d6de56fd: Waiting
d85d1128dd96: Waiting
4ad5ea901fd1: Waiting
37892ffbfcaa: Waiting
1f3e46996e29: Waiting
failed to authorize: failed to fetch oauth token: unexpected status from GET request to https://myjavaapp.azurecr.io/v2/_auth/login?service=acr&service=https://myjavaapp.azurecr.io: 401 Unauthorized
PS E:\projects\terraformProj\nodeWebApp> az acr login --name myjavaapp
Login Succeeded
PS E:\projects\terraformProj\nodeWebApp> docker push myjavaapp.azurecr.io/node-web-app:v1
The push refers to repository [myjavaapp.azurecr.io/node-web-app]
5650d6de56fd: Pushed
c1875295cf1f: Pushed
1f3e46996e29: Pushed
d85d1128dd96: Pushed
4ad5ea901fd1: Pushed
37892ffbfcaa: Pushed
6504e29600c8: Pushed
v1: digest: sha256:43242ddf49e080d7dd55b603fe355155885581e1a807c54a14b6bc252889e12d size: 856
```

```

1 apiVersion: apps/v1
2 kind: Deployment
3 metadata:
4 name: node-web-app
5 spec:
6 replicas: 2
7 selector:
8 matchLabels:
9 app: node-web-app
10 template:
11 metadata:
12 labels:
13 app: node-web-app
14 spec:
15 containers:
16 - name: node-web-app
17 image: myjavaapp.azurecr.io/node-web-app:v1
18 ports:
19 - containerPort: 3000
20

```

Microsoft Azure

myjavaapp | Repositories

Home > myjavaapp > mywebdemo > node-web-app

Search resources, services, and docs (S+)

Activity log Access control (IAM) Quick start Events Settings Services Repositories Webhooks Geo-replications Tasks Connected registries (Preview)

Search Refresh Manage Deleted Repositories

How to use ACR. Artifact streaming helps pull images faster from AKS clusters. The Artifact streaming status column shows which repositories are using this feature. [Learn more](#)

Search to filter repositories ...

Repositories Cache Rule

mywebdemo node-web-app

Microsoft Azure

node-web-app | Overview

Deployment

Search Refresh View Grafana workspaces Give feedback Configure

Essentials Namespace: default Ready: 2/2 Revision history: 10 Strategy type: RollingUpdate Labels: Selector: app=node-web-app

Cluster: alka-default Creation time: 2021-01-24T15:04:23.002Z Min ready seconds: 0 Replicas: 2 desired, 2 updated, 2 total, 2 available, 0 unavailable Rolling update strategy: 25% max unavailable, 25% max surge See more

Events

| Total | Warning | Normal |
|-------|---------|--------|
| 2     | 0       | 2      |

Workloads

Pods Replica sets

| Pod name                    | Ready | Status  | Restart count | Age        | CPU      | Memory   | Node                | Pod IP       |
|-----------------------------|-------|---------|---------------|------------|----------|----------|---------------------|--------------|
| node-web-app-d876559db-4c5q | 1/1   | Running | 0             | 12 minutes | 0m 24.0m | 0m 24.0m | alka-default-227... | 10.244.0.117 |
| node-web-app-d876559db-7g7o | 1/1   | Running | 0             | 12 minutes | 0m 24.0m | 0m 24.0m | alka-default-227... | 10.244.0.117 |

Resource utilization

```
PS E:\projects\terraformProj\nodeWebApp> kubectl cluster-info
Kubernetes control plane is running at https://myaksdns-7txr3evi.hcp.westeurope.azurek8s.io:443
CoreDNS is running at https://myaksdns-7txr3evi.hcp.westeurope.azurek8s.io:443/api/v1/namespaces/kube-system/services/kube-dns:dns/proxy
Metrics-server is running at https://myaksdns-7txr3evi.hcp.westeurope.azurek8s.io:443/api/v1/namespaces/kube-system/services/https:metrics-server:/proxy

To further debug and diagnose cluster problems, use 'kubectl cluster-info dump'.
PS E:\projects\terraformProj\nodeWebApp> az acr repository list --name myjavaapp --output table
Result
mywebdemo
node-web-app
PS E:\projects\terraformProj\nodeWebApp> kubectl apply -f node-web-app-deployment.yaml
deployment.apps/node-web-app created
PS E:\projects\terraformProj\nodeWebApp> kubectl apply -f node-web-app-service.yaml
service/node-web-app-service created
PS E:\projects\terraformProj\nodeWebApp> kubectl get deployments
NAME READY UP-TO-DATE AVAILABLE AGE
node-web-app 0/2 2 0 18s
PS E:\projects\terraformProj\nodeWebApp> kubectl get pods
NAME READY STATUS RESTARTS AGE
node-web-app-d876558db-z5nnk 0/1 ImagePullBackOff 0 25s
node-web-app-d876558db-zrdgg 0/1 ImagePullBackOff 0 25s
PS E:\projects\terraformProj\nodeWebApp> kubectl get services
NAME TYPE CLUSTER-IP EXTERNAL-IP PORT(S) AGE
kubernetes ClusterIP 10.0.0.1 <none> 443/TCP 11m
node-web-app-service LoadBalancer 10.0.115.85 57.153.38.238 80:31347/TCP 25s
PS E:\projects\terraformProj\nodeWebApp> kubectl get svc node-web-app-service
NAME TYPE CLUSTER-IP EXTERNAL-IP PORT(S) AGE
node-web-app-service LoadBalancer 10.0.115.85 57.153.38.238 80:31347/TCP 39s
PS E:\projects\terraformProj\nodeWebApp> |
```



Hello from Node.js app on AKS!

## Practical Task 7: Configure and Use ConfigMaps and Secrets in AKS

### Requirements:

1. Create a ConfigMap to store non-sensitive configuration data with only the required key value pairs for the application.
2. Create a Kubernetes Secret to store sensitive data (e.g., API keys) with the least amount of information needed.
3. Update the application deployment to use the ConfigMap and Secret.
4. Remove the ConfigMap, Secret, and deployment after testing.

## Actions Taken:

```

apiVersion: v1
kind: Secret
metadata:
 name: my-app-secret
 namespace: default
 type: Opaque
data:
 DB_PASSWORD: d812aQ==
 API_TOKEN: c2VjcmV0dG9rZW4=

```

```

Windows PowerShell x + -
-a----- 1/26/2025 7:48 PM 961 node-web-app-deployment.yaml
-a----- 1/26/2025 4:35 PM 208 node-web-app-service.yaml
-a----- 1/26/2025 4:08 PM 326 server.js

PS E:\projects\terraformProj\nodeWebApp> kubectl apply -f configmap.yaml
configmap/my-app-config created
PS E:\projects\terraformProj\nodeWebApp> kubectl get configmap my-app-config -o yaml
apiVersion: v1
data:
 API_URL: https://api.myapp.com
 APP_NAME: MyAKSApp
 LOG_LEVEL: info
kind: ConfigMap
metadata:
 annotations:
 kubectl.kubernetes.io/last-applied-configuration:
 {"apiVersion":"v1","data":{"API_URL":"https://api.myapp.com","APP_NAME":"MyAKSApp","LOG_LEVEL":"info"},"kind":"ConfigMap"}
 , "metadata": {"annotations": {}, "name": "my-app-config", "namespace": "default"}
 creationTimestamp: "2025-01-26T19:16:22Z"
 name: my-app-config
 namespace: default
 resourceVersion: 32389
 uid: de7ccc04-93d7-41fe-0301102c7b05
PS E:\projects\terraformProj\nodeWebApp> [Convert]::ToBase64String([Text.Encoding]::UTF8.GetBytes("vivi"))
dn12aQ==
PS E:\projects\terraformProj\nodeWebApp> [Convert]::ToBase64String([Text.Encoding]::UTF8.GetBytes("secrettoken"))
c2VjcmV0dG9rZW4=
PS E:\projects\terraformProj\nodeWebApp>

```

```
PS E:\projects\terraformProj\terraform-aks> kubectl create configmap app-config --from-literal=APP_NAME="Node Web Application" --from-literal=APP_ENV="production" --from-literal=LOG_LEVEL="info"
```

```
PS E:\projects\terraformProj\terraform-aks> kubectl create secret generic app-secret --from-literal=DB_PASSWORD="SuperSecretPassword" --from-literal=API_KEY="12345-ABCDE"
```

```

10
11 resource "azurerm_kubernetes_cluster" "aks" {
12 name = var.aks_cluster_name
13 location = "westeurope"
14 resource_group_name = "Volodymyr-Dibrova"
15 dns_prefix = "myaksdns"
16
17 default_node_pool {
18 name = "default"
19 node_count = var.node_count
20 vm_size = var.vm_size
21 }
22
23 identity {
24 type = "SystemAssigned"
25 }
26
27 tags = {
28 environment = "dev"
29 }
30
31
32 output "kube_config" {
33 value = azurerm_kubernetes_cluster.aks.kube_config_raw
34 sensitive = true
35 }
36
37
38 resource "kubernetes_config_map" "app_config" {
39 metadata {
40 name = "app-config"
41 namespace = "default"
42 }
43
44 data = [
45 APP_NAME = "Node Web Application"
46 APP_ENV = "production"
47 LOG_LEVEL = "info"
48]
49 }
50
51 resource "kubernetes_secret" "app_secret" {
52 metadata {
53 name = "app-secret"
54 namespace = "default"
55 }
56
57 data = [
58 DB_PASSWORD = "SuperSecretPassword"
59 API_KEY = "12345-ABCDE"
60]
61
62 type = "Opaque"
63
64 }
65

```

Microsoft Azure

Home > myakscuster | Configuration >

**app-secret** | Overview ...

Secret

Search Refresh Give feedback

| Essential |           |               |                            |
|-----------|-----------|---------------|----------------------------|
| Namespace | : default | Cluster       | : myakscuster              |
| Labels    | : -       | Creation time | : 2025-01-26T20:30:02.000Z |
| See more  |           | Type          | : Opaque                   |

Data

|             |                             |
|-------------|-----------------------------|
| API_KEY     | MtIzNDU0QUJDREU=            |
| DB_PASSWORD | USVwZXUTZWlyZkRQYWNzdz9ZA== |

Screenshot of a Microsoft Azure DevOps pipeline interface showing multiple tabs open (configmap.yaml, secret.yaml, deployment.yaml, main.tf, variables.tf, node-web-app-deployment.yaml) and a code editor displaying the deployment configuration.

```

1 apiVersion: apps/v1
2 kind: Deployment
3 metadata:
4 name: node-web-app
5 spec:
6 replicas: 2
7 selector:
8 matchLabels:
9 app: node-web-app
10 template:
11 metadata:
12 labels:
13 app: node-web-app
14 spec:
15 containers:
16 - name: node-web-app
17 image: myjavaapp.azurecr.io/node-web-app:v1
18 ports:
19 - containerPort: 8080
20 env:
21 - name: APP_NAME
22 valueFrom:
23 configMapKeyRef:
24 name: app-config
25 key: APP_NAME
26 - name: DB_PASSWORD
27 valueFrom:
28 secretKeyRef:
29 name: app-secret
30 key: DB_PASSWORD
31

```

Below the code editor is the Microsoft Azure portal "node-web-app | Overview" page. It shows the deployment status (2 ready, 2 total), logs, events (2 total, 0 warnings, 2 normal), and pods (2 running). A terminal window at the bottom shows the command-line output of the deployment process.

```

Apply complete! Resources: 1 added, 1 changed, 0 destroyed.

Outputs:
aks_fqdn = "myaksdns-n72ja54b.hcp.westeurope.azmk8s.io"
kube_config = <sensitive>
PS E:\projects\terraformProj\terraform-aks> kubectl get deployments
NAME READY UP-TO-DATE AVAILABLE AGE
node-web-app 2/2 2 2 23s
PS E:\projects\terraformProj\terraform-aks> kubectl logs deployment/node-web-app
Found 2 pods, using pod/node-web-app-57596f7f98-vpwqd
Server is running on port 3000
PS E:\projects\terraformProj\terraform-aks> kubectl get services
NAME TYPE CLUSTER-IP EXTERNAL-IP PORT(S) AGE
kubernetes ClusterIP 10.0.0.1 <none> 443/TCP 3h54m
PS E:\projects\terraformProj\terraform-aks> kubectl get pods
NAME READY STATUS RESTARTS AGE
node-web-app-57596f7f98-vpwqd 1/1 Running 0 11m
node-web-app-57596f7f98-xvtjj 1/1 Running 0 11m
PS E:\projects\terraformProj\terraform-aks> kubectl exec -it node-web-app-57596f7f98-vpwqd -- printenv
PATH=/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin
HOSTNAME=node-web-app-57596f7f98-vpwqd
NODE_VERSION=18.20.6
YARN_VERSION=1.22.22
APP_NAME=Node Web Application
DB_PASSWORD=SuperSecretPassword
KUBERNETES_SERVICE_PORT_HTTPS=443
KUBERNETES_PORT=tcp://10.0.0.1:443
KUBERNETES_PORT_443_TCP=tcp://10.0.0.1:443
KUBERNETES_PORT_443_TCP_PROTO=tcp
KUBERNETES_PORT_443_TCP_PORT=443
KUBERNETES_PORT_443_TCP_ADDR=10.0.0.1
KUBERNETES_SERVICE_HOST=10.0.0.1
KUBERNETES_SERVICE_PORT=443
TERM=xterm
HOME=/root
PS E:\projects\terraformProj\terraform-aks> |

```

node-web-app | Overview

**Essentials**

- Namespace: default
- Ready: 2/2
- Revision history: 10
- Strategy type: RollingUpdate
- Labels: app=node-web-app
- Selector: app=node-web-app

**Events**

| Total | Warning | Normal |
|-------|---------|--------|
| 2     | 0       | 2      |

**Workloads**

| Pods                                                  | Replica sets                            | Events                                      | Show labels                            |                                     |                                                      |                                       |                                             |                                       |
|-------------------------------------------------------|-----------------------------------------|---------------------------------------------|----------------------------------------|-------------------------------------|------------------------------------------------------|---------------------------------------|---------------------------------------------|---------------------------------------|
| <input type="checkbox"/> Pod name                     | <input type="checkbox"/> Ready          | <input type="checkbox"/> Status             | <input type="checkbox"/> Restart count | <input type="checkbox"/> Age        | <input type="checkbox"/> CPU                         | <input type="checkbox"/> Memory       | <input type="checkbox"/> Node               | <input type="checkbox"/> Pod IP       |
| <input type="checkbox"/> node-web-app-371997798-rpved | <input checked="" type="checkbox"/> 1/1 | <input checked="" type="checkbox"/> Running | <input type="checkbox"/> 0             | <input type="checkbox"/> 24 minutes | <input checked="" type="checkbox"/> 0 enable metrics | <input type="checkbox"/> 0            | <input type="checkbox"/> aks-default-242... | <input type="checkbox"/> 10.244.0.95  |
| <input type="checkbox"/> node-web-app-371997798-mqg   | <input checked="" type="checkbox"/> 1/1 | <input checked="" type="checkbox"/> Running | <input type="checkbox"/> 0             | <input type="checkbox"/> 24 minutes | <input checked="" type="checkbox"/> 0                | <input checked="" type="checkbox"/> 0 | <input type="checkbox"/> aks-default-242... | <input type="checkbox"/> 10.244.0.153 |

**Resource utilization**

## Practical Task 8: Scale Applications in AKS

### Requirements:

1. Deploy a stateless application to the AKS cluster using minimal resource specifications.
2. Use the `kubectl scale` command to manually scale the application to only 2–3 replicas for testing.
3. Set up Horizontal Pod Autoscaler (HPA) with reasonable CPU usage thresholds to minimize pod creation.
4. Simulate load on the application for a short duration and remove the deployment after observing the scaling behavior.

```

1 apiVersion: apps/v1
2 kind: Deployment
3
4 metadata:
5 name: stateless-app
6 labels:
7 app: stateless-app
8
9 spec:
10 replicas: 1
11 selector:
12 matchLabels:
13 app: stateless-app
14 template:
15 metadata:
16 labels:
17 app: stateless-app
18 spec:
19 containers:
20 - name: stateless-app
21 image: nginx
22 resources:
23 requests:
24 cpu: "250m"
25 memory: "128Mi"
26 limits:
27 cpu: "500m"
28 memory: "256Mi"
29 ports:
30 - containerPort: 80

```

```

1 apiVersion: v1
2 kind: Service
3
4 metadata:
5 name: stateless-app
6
7 spec:
8 selector:
9 app: stateless-app
10 ports:
11 - protocol: TCP
12 port: 80
13 targetPort: 80

```

Microsoft Azure

Home > Kubernetes services > myakscluster | Workloads >

### stateless-app | Overview

Deployment

Search Refresh View Grafana workspaces Give feedback

**Overview**

- YAML
- Events
- Live logs

**Essentials**

|                  |                     |                         |                                                             |
|------------------|---------------------|-------------------------|-------------------------------------------------------------|
| Namespace        | : default           | Cluster                 | : myakscluster                                              |
| Ready            | : 1/1               | Creation time           | : 2025-01-27T18:43:17.000Z                                  |
| Revision history | : 10                | Min ready seconds       | : 0                                                         |
| Strategy type    | : RollingUpdate     | Replicas                | : 1 desired, 1 updated, 1 total, 1 available, 0 unavailable |
| Labels           | : app=stateless-app | Rolling update strategy | : 25% max unavailable, 25% max surge                        |
| Selector         | : app=stateless-app | See more                |                                                             |

**Events**

| Total | Warning | Normal |
|-------|---------|--------|
| 0     | 0       | 0      |

**Resource utilization**

**Workloads**

**Pods**

| Pod name                     | Ready | Status  | Restart count | Age      | CPU | Memory | Node               | Pod IP      |
|------------------------------|-------|---------|---------------|----------|-----|--------|--------------------|-------------|
| stateless-app-79c79dd5-427r7 | 1/1   | Running | 0             | 13 hours | 8%  | 1%     | aks-default-188... | 10.244.0.62 |

```

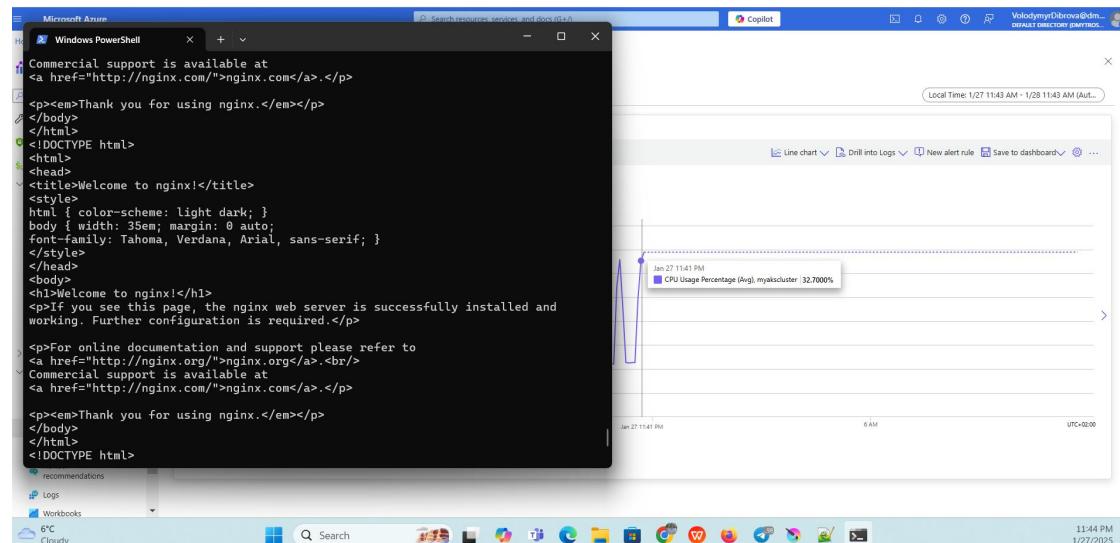
PS E:\projects\terraformProj\nodeWebApp\stateless> kubectl apply -f stateless-app.yaml
deployment.apps/stateless-app created
PS E:\projects\terraformProj\nodeWebApp\stateless> kubectl get deployments
NAME READY UP-TO-DATE AVAILABLE AGE
stateless-app 1/1 1 1 17m
PS E:\projects\terraformProj\nodeWebApp\stateless> kubectl get deployment stateless-app
NAME READY UP-TO-DATE AVAILABLE AGE
stateless-app 1/1 1 1 19m
PS E:\projects\terraformProj\nodeWebApp\stateless> kubectl scale deployment stateless-app --replicas=3
deployment.apps/stateless-app scaled
PS E:\projects\terraformProj\nodeWebApp\stateless> kubectl get deployment stateless-app
NAME READY UP-TO-DATE AVAILABLE AGE
stateless-app 3/3 3 3 20m
PS E:\projects\terraformProj\nodeWebApp\stateless> kubectl get pods -l app=stateless-app
NAME READY STATUS RESTARTS AGE
stateless-app-79c79dd5-5chgz 1/1 Running 0 31s
stateless-app-79c79dd5-c6g2j 1/1 Running 0 31s
stateless-app-79c79dd5-hl67q 1/1 Running 0 20m
PS E:\projects\terraformProj\nodeWebApp\stateless>

```

```

PS E:\projects\terraformProj\nodeWebApp\stateless> kubectl get pods -l app=stateless-app
NAME READY STATUS RESTARTS AGE
stateless-app-79c79dd5-5chgz 1/1 Running 0 31s
stateless-app-79c79dd5-c6g2j 1/1 Running 0 31s
stateless-app-79c79dd5-hl67q 1/1 Running 0 20m
PS E:\projects\terraformProj\nodeWebApp\stateless> kubectl autoscale deployment stateless-app --cpu-percent=50 --min=1 --max=5
horizontalpodautoscaler.autoscaling/stateless-app autoscaled
PS E:\projects\terraformProj\nodeWebApp\stateless> kubectl get hpa
NAME REFERENCE TARGETS MINPODS MAXPODS REPLICAS AGE
stateless-app Deployment/stateless-app cpu: 0%/50% 1 5 3 22s
PS E:\projects\terraformProj\nodeWebApp\stateless> kubectl describe hpa stateless-app
Name: stateless-app
Namespace: default
Labels: <none>
Annotations: <none>
CreationTimestamp: Mon, 27 Jan 2025 21:07:44 +0200
Reference: Deployment/stateless-app
Metrics: current / target
 resource cpu on pods (as a percentage of request): 0% (0) / 50%
Min replicas: 1
Max replicas: 5
Deployment pods: 3 current / 3 desired
Conditions:
 Type Status Reason Message
 ---- ---- ---- -----
 AbleToScale True ScaledDownStabilized recent recommendations were higher than current one, applying the highest recent recommendation
 ScalingActive True ValidMetricFound the HPA was able to successfully calculate a replica count from cpu resource utilization (percentage of requested CPU)
st)

```



```

PS E:\projects\terraformProj\nodeWebApp\stateless> kubectl get hpa -w
NAME REFERENCE TARGETS MINPODS MAXPODS REPLICAS AGE
stateless-app Deployment/stateless-app cpu: 11%/50% 1 5 1 159m
|
```

## load generation on pods

```

PS C:\Users\User> kubectl run -i --tty load-generator --image=busybox -- /bin/sh -c "while true; do wget -q -O- http://stateless-app.default.svc.cluster.local; done"

```

## Autoscaler monitoring

```
PS C:\Users\User> kubectl get hpa -w
```

The screenshot shows two adjacent Windows PowerShell windows. The left window displays the raw HTML response from an Nginx deployment, indicating a successful connection. The right window shows the command-line output of 'kubectl get hpa -w' for a deployment named 'stateless-app'. The output lists three pods: one at 10% CPU usage (labeled '16m'), one at 11% CPU usage (labeled '16m'), and one at 166% CPU usage (labeled '166m'). The final command in the right window is 'PS E:\projects\terraformProj\nodeWebApp\stateless> kubectl get hpa -w'.

```
<p>Thank you for using nginx!</p>
</html>
<!DOCTYPE html>
<html>
<head>
<title>Welcome to nginx!</title>
<style>
html { color-scheme: light dark; }
body { width: 35em; margin: 0 auto;
font-family: Tahoma, Verdana, Arial, sans-serif; }
</style>
</head>
<body>
<h1>Welcome to nginx!</h1>
<p>If you see this page, the nginx web server is successfully installed!
working. Further configuration is required.</p>
<p>For online documentation and support please refer to
nginx.org.

Commercial support is available at
nginx.com.</p>
<p>Thank you for using nginx!</p>
</body>
</html>
<!DOCTYPE html>
<html>
<head>
<title>Welcome to nginx!</title>
<style>
html { color-scheme: light dark; }
body { width: 35em; margin: 0 auto;
font-family: Tahoma, Verdana, Arial, sans-serif; }
</style>
</head>
<body>
<h1>Welcome to nginx!</h1>
<p>If you see this page, the nginx web server is successfully installed!
working. Further configuration is required.</p>
```

NAME	READY	STATUS	RESTARTS	AGE
stateless-app-79c79dd5-5chgz	1/1	Running	0	7s
stateless-app-79c79dd5-427rt	1/1	Running	0	7s
stateless-app-79c79dd5-5chgz	1/1	Running	0	166m

```
PS E:\projects\terraformProj\nodeWebApp\stateless> kubectl scale deployment stateless-app --replicas=3
PS E:\projects\terraformProj\nodeWebApp\stateless> kubectl get pods -l app=stateless-app
NAME READY STATUS RESTARTS AGE
stateless-app-79c79dd5-25d2n 1/1 Running 0 7s
stateless-app-79c79dd5-427rt 1/1 Running 0 7s
stateless-app-79c79dd5-5chgz 1/1 Running 0 166m
PS E:\projects\terraformProj\nodeWebApp\stateless> kubectl get hpa -w
NAME REFERENCE TARGETS MINPODS MAXPODS REP
LICAS AGE
stateless-app Deployment/stateless-app cpu: 6%/50% 1 5 3
162m
PS E:\projects\terraformProj\nodeWebApp\stateless> kubectl get hpa -w
NAME REFERENCE TARGETS MINPODS MAXPODS REP
LICAS AGE
stateless-app Deployment/stateless-app cpu: 6%/50% 1 5 3
163m
PS E:\projects\terraformProj\nodeWebApp\stateless> kubectl get hpa -w
NAME REFERENCE TARGETS MINPODS MAXPODS REP
LICAS AGE
stateless-app Deployment/stateless-app cpu: 3%/50% 1 5 3
163m
PS E:\projects\terraformProj\nodeWebApp\stateless> kubectl get hpa -w
```

## Practical Task 9: Rolling Update of an Application in AKS

### Requirements:

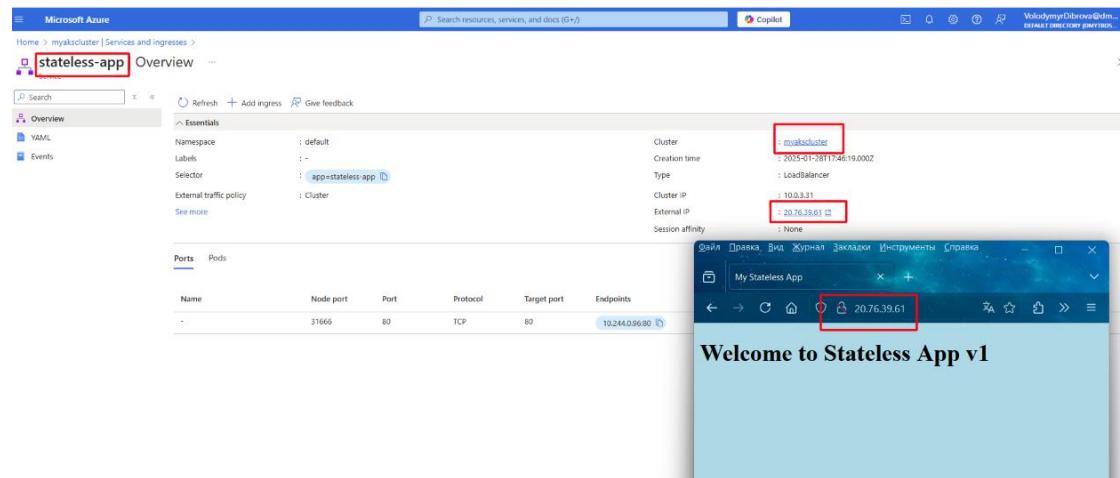
1. Deploy a lightweight version of your application to the AKS cluster.
2. Update the Docker image to a new version with minimal changes (e.g., color change).
3. Perform a rolling update using `kubectl set image` with minimal replicas to reduce resource usage.
4. Verify the update process quickly and remove the deployment after the update.

1.Created a YAML file for deployment, service, and docker image with v1

The screenshot shows the Microsoft Azure Container Registry interface for the 'myjavaapp' repository. The 'stateless-app' service is selected, highlighted with a red box. The preview pane displays the following HTML code:

```
<!DOCTYPE html>
<html>
<head>
 <title>My Stateless App</title>
 <style>
 body { background-color: lightblue; }
 </style>
</head>
<body>
 <h1>Welcome to Stateless App v1</h1>
</body>
</html>
```

2.Launched the service and opened external-IP in bowser



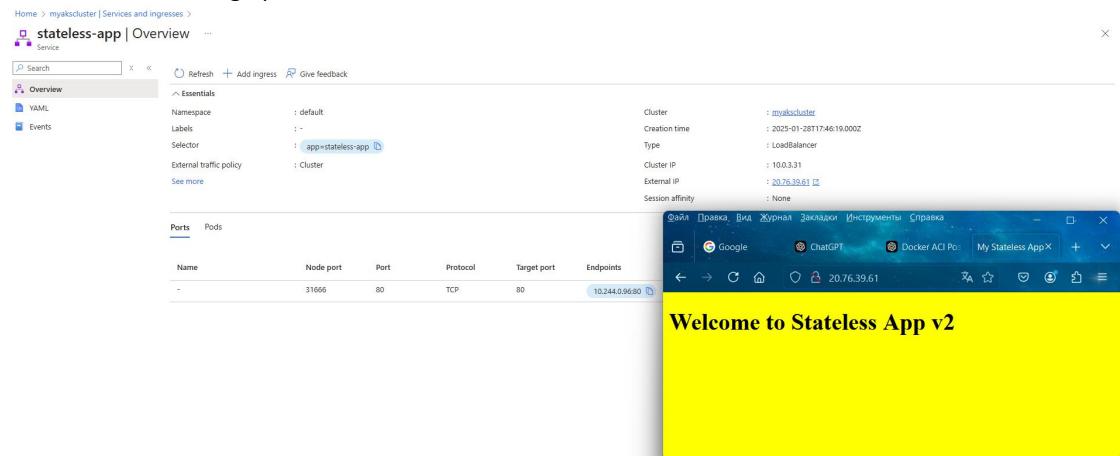
3. Created a new version of the image (v2) with minimal changes - updated body color to yellow

```

1 <!DOCTYPE html>
2 <html>
3 <head>
4 <title>My Stateless App</title>
5 <style>
6 body { background-color: yellow; }
7 </style>
8 </head>
9 <body>
10 <h1>Welcome to Stateless App v2</h1>
11 </body>
12 </html>
13

```

4. Performed a rolling update



```

Windows PowerShell x + v
kubernetes ClusterIP 10.0.3.1 <none> 443/TCP 57m
stateless-app LoadBalancer 10.0.3.31 28.76.39.61 80:31666/TCP 39m
PS E:\projects\terraform\pro\nodevwebapp\Rolling- kubectl describe deployment stateless-app
Name: stateless-app
Namespace: default
Labels: app=stateless-app
Annotations: deployment.kubernetes.io/revision: 4
 kubernetes.io/change-cause: kubectl.exe set image deployment/stateless-app stateless-app=myjavaapp.azurecr.io/stateless-app:v2 --record=true
Selector: app=stateless-app
Replicas: 1 desired | 1 updated | 1 total | 1 available | 0 unavailable
StrategyType: RollingUpdate
MinReadySeconds: 0
RollingUpdateStrategy: 25% max unavailable, 25% max surge
Pod Template:
 Labels: app=stateless-app
 Annotations: kubectl.kubernetes.io/restartedAt: 2025-01-28T20:06:11+02:00
 Containers:
 stateless-app:
 Image: myjavaapp.azurecr.io/stateless-app:v2
 Port: 80/TCP
 Host Port: 8/TCP
 Limits:
 cpu: 500m
 memory: 256Mi
 Requests:
 cpu: 250m
 memory: 128Mi
 Environment: <none>
 Mounts: <none>
 Volumes: <none>
 Node-selectors: <none>
 Tolerations: <none>
 Conditions:
 Type Status Reason
 Available True MinimumReplicasAvailable
 Progressing True NewReplicaSetAvailable
OldReplicaSets: stateless-app-79c79dd5 (0/0 replicas created), stateless-app-5886cc7598 (0/0 replicas created), stateless-app-7f95cd4fb8 (0/0 replicas created)
NewReplicaSet: stateless-app-85bd98cc85 (1/1 replicas created)
Events:
 Type Reason Age From Message
 Normal ScalingReplicaSet 43m deployment-controller Scaled up replica set stateless-app-79c79dd5 to 1
 Normal ScalingReplicaSet 24m deployment-controller Scaled up replica set stateless-app-5886cc7598 to 1
 Normal ScalingReplicaSet 23m deployment-controller Scaled down replica set stateless-app-5886cc7598 to 0 from 1
 Normal ScalingReplicaSet 23m deployment-controller Scaled up replica set stateless-app-7f95cd4fb8 to 1 from 0
 Normal ScalingReplicaSet 16m deployment-controller Scaled down replica set stateless-app-79c79dd5 to 0 from 1
 Normal ScalingReplicaSet 3m47s deployment-controller Scaled up replica set stateless-app-85bd98cc85 to 1
 Normal ScalingReplicaSet 3m43s deployment-controller Scaled down replica set stateless-app-7f95cd4fb8 to 0 from 1

```

## 10:Bonus task. **GitOps with AKS Requirements:**

1. Setup ArgoCD on Azure Kubernetes Services
2. Perform image update on cluster

### Actions Taken:

1. Created a namespace for ArgoCD and Installing ArgoCD in AKS

```

PS E:\projects\terraformProj\terraform-aks> az aks get-credentials --resource-group Volodymyr-Dibrova --name myakscluster
Merged "myakscluster" as current context in C:\Users\sergey\kubeconfig
PS E:\projects\terraformProj\terraform-aks> kubectl get nodes
NAME STATUS ROLES AGE VERSION
aks-default-28313743-vmsss000000 Ready <none> 16h v1.30.7
aks-default-28313743-vmsss000001 Ready <none> 16h v1.30.7
PS E:\projects\terraformProj\terraform-aks> kubectl create namespace argocd
namespace/argocd created
PS E:\projects\terraformProj\terraform-aks> kubectl apply -n argocd --ff https://raw.githubusercontent.com/argoproj/argo-cd/stable/manifests/install.yaml
customresourcedefinition.apiextensions.k8s.io/applications.argoproj.io created
customresourcedefinition.apiextensions.k8s.io/applicationsets.argoproj.io created
customresourcedefinition.apiextensions.k8s.io/appprojects.argoproj.io created
serviceaccount/argocd-application-controller created
serviceaccount/argocd-applicationset-controller created
serviceaccount/argocd-dex-server created
serviceaccount/argocd-notifications-controller created
serviceaccount/argocd-redis created
serviceaccount/argocd-repo-server created
serviceaccount/argocd-server created
role.rbac.authorization.k8s.io/argocd-application-controller created
role.rbac.authorization.k8s.io/argocd-applicationset-controller created
role.rbac.authorization.k8s.io/argocd-dex-server created
role.rbac.authorization.k8s.io/argocd-notifications-controller created
role.rbac.authorization.k8s.io/argocd-redis created
role.rbac.authorization.k8s.io/argocd-server created
clusterrole.rbac.authorization.k8s.io/argocd-application-controller created
clusterrole.rbac.authorization.k8s.io/argocd-applicationset-controller created
rolebinding.rbac.authorization.k8s.io/argocd-application-controller created
rolebinding.rbac.authorization.k8s.io/argocd-dex-server created
rolebinding.rbac.authorization.k8s.io/argocd-notifications-controller created
rolebinding.rbac.authorization.k8s.io/argocd-redis created
rolebinding.rbac.authorization.k8s.io/argocd-server created
clusterrolebinding.rbac.authorization.k8s.io/argocd-application-controller created
clusterrolebinding.rbac.authorization.k8s.io/argocd-applicationset-controller created
clusterrolebinding.rbac.authorization.k8s.io/argocd-server created
configmap/argocd-cm created
configmap/argocd-cmd-params-cm created
configmap/argocd-gpg-keys-cm created
configmap/argocd-notifications-cm created

```

## 2.Temporary access via port-forward

Windows PowerShell

```

Error from server (BadRequest): error decoding JSON response
Error from server (NotFound): services "argo-cd"
PS E:\projects\terraformProj\terraform-als> kubectl get svc
Error from server (BadRequest): error decoding JSON response
PS E:\projects\terraformProj\terraform-als> kubectl delete svc "argo-cd-server" deleted
PS E:\projects\terraformProj\terraform-als> kubectl expose deployment argocd-server --name=argocd-server -n argocd --type=LoadBalancer --port=443 --target-p
ort=8080
service/argocd-server exposed
PS E:\projects\terraformProj\terraform-als> kubectl get svc
NAME TYPE CLUSTER-IP EXTERNAL-IP PORT(S) AGE
argocd-server LoadBalancer 10.0.7.158 <pending> 443:31772/TCP 7s
PS E:\projects\terraformProj\terraform-als> kubectl get svc -n argocd argocd-server
NAME TYPE CLUSTER-IP EXTERNAL-IP PORT(S) AGE
argocd-server LoadBalancer 10.0.7.158 51.138.42.205 443:31772/TCP 14s
PS E:\projects\terraformProj\terraform-als> kubectl port-forward svc/argocd-server -n argocd 8080:443
Forwarding from 127.0.0.1:8080 -> 8080
Forwarding from [:1]:8080 -> 8080
Handling connection for 8080
E0129 12:56:51.932264 15232 portforward.go:385] error copying from remote stream to local connection: readfrom tcp4 127.0.0.1:8080->127.0.0.1:64770: write
tcp4 127.0.0.1:8080->127.0.0.1:64770: wsasend: An established connection was aborted by the software in your host machine.
PS E:\projects\terraformProj\terraform-als> *C
PS E:\projects\terraformProj\terraform-als> kubectl port-forward svc/argocd-server -n argocd 8080:443
Forwarding from 127.0.0.1:8080 -> 8080
Forwarding from [:1]:8080 -> 8080
Handling connection for 8080
Handling connection for 8080

```

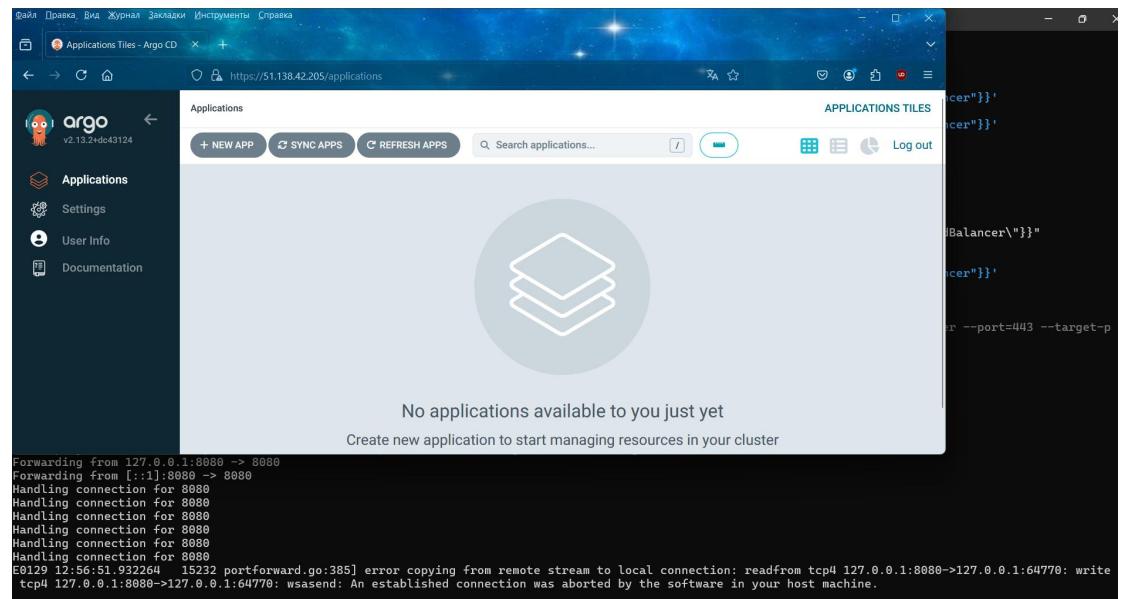
## 3.Deleted the old service and created new service LoadBalancer

```

PS E:\projects\terraformProj\terraform-aks> kubectl delete svc argocd-server -n argocd
service "argocd-server" deleted
PS E:\projects\terraformProj\terraform-aks> kubectl expose deployment argocd-server --name=argocd-server -n argocd --type=LoadBalancer --port=443 --target-p
ort=8080
service/argocd-server exposed
PS E:\projects\terraformProj\terraform-aks> kubectl get svc -n argocd argocd-server
NAME TYPE CLUSTER-IP EXTERNAL-IP PORT(S) AGE
argocd-server LoadBalancer 10.0.7.158 <pending> 443:31772/TCP 7s
PS E:\projects\terraformProj\terraform-aks> kubectl get svc -n argocd argocd-server
NAME TYPE CLUSTER-IP EXTERNAL-IP PORT(S) AGE
argocd-server LoadBalancer 10.0.7.158 51.138.42.205 443:31772/TCP 14s
PS E:\projects\terraformProj\terraform-aks> kubectl port-forward svc/argocd-server -n argocd 8080:443
Forwarding from 127.0.0.1:8080 -> 8080
Forwarding from [:1]:8080 -> 8080
Handling connection for 8080
E0129 12:56:51.932264 15232 portforward.go:385] error copying from remote stream to local connection: readfrom tcp4 127.0.0.1:8080->127.0.0.1:64770: write
tcp4 127.0.0.1:8080->127.0.0.1:64770: wsasend: An established connection was aborted by the software in your host machine.

```

## 4. Open with external ip:



## 5. connect CLI to ArgoCD server

```

PS E:\projects\terraformProj\nodeWebApp\Rolling> argocd version
argocd: v2.13.3+ha25c8a0
 BuildDate: 2025-01-03T19:06:52Z
 GitCommit: a25c8a0eef7830be0c2c9074c92dbea8ff23a962
 GitTreeState: clean
 GoVersion: go1.22.10
 Compiler: gc
 Platform: windows/amd64
time="2025-01-29T13:24:46+02:00" level=fatal msg="Argo CD server address unspecified"
PS E:\projects\terraformProj\nodeWebApp\Rolling> argocd login 51.138.42.205 --username admin --password -Qa3rijL0fPrKbRH --insecure
'admin:login' logged in successfully
Context '51.138.42.205' updated
PS E:\projects\terraformProj\nodeWebApp\Rolling> |

```

Name	Node port	Port	Protocol	Target port	Endpoints
-	31772	443	TCP	8080	10.244.0.156:8080

## 6. synch in gitops

```

PS E:\projects\terraformProj\nodeWebApp\Rolling> argocd app create my-app `
>> --repo https://github.com/Vivien87/mywebdemo.git `
>> --path . `
>> --dest-server https://kubernetes.default.svc `
>> --dest-namespace default `
application 'my-app' created
PS E:\projects\terraformProj\nodeWebApp\Rolling> argocd app get my-app
Name: argocd/my-app
Project: default
Server: https://kubernetes.default.svc
Namespace: default
URL: https://51.138.42.205/applications/my-app
Source:
- Repo:
 Target: https://github.com/Vivien87/mywebdemo.git
 Path: .
SyncWindow: Sync Allowed
Sync Policy: Manual
Sync Status: Synced to (9d82e02)
Health Status: Healthy
PS E:\projects\terraformProj\nodeWebApp\Rolling> argocd app sync my-app

Name: argocd/my-app
Project: default
Server: https://kubernetes.default.svc
Namespace: default
URL: https://51.138.42.205/applications/my-app
Source:
- Repo:
 Target: https://github.com/Vivien87/mywebdemo.git
 Path: .
SyncWindow: Sync Allowed
Sync Policy: Manual
Sync Status: Synced to (9d82e02)
Health Status: Healthy

Operation: Sync
Sync Revision: 9d82e02e03a4c6bf26b2f763d11ee3bd4dba1979
Phase: Succeeded
Start: 2025-01-29 13:37:11 +0200 EET
Finished: 2025-01-29 13:37:11 +0200 EET
Duration: 0s
Message: successfully synced (no more tasks)
PS E:\projects\terraformProj\nodeWebApp\Rolling> |

```

The screenshot shows the Argo CD web interface with the following details:

- Project:** default
- Status:** healthy
- Repository:** https://github.com/Vivien87/mywebdem...
- Target Ref:** HEAD
- Path:** .
- Destination:** in-cluster
- Namespace:** default
- Created:** 01/29/2025 13:36:14 (9 minutes ago)
- Last Sync:** 01/29/2025 13:37:11 (8 minutes ago)

Below the application card, there are three buttons: SYNC, REFRESH, and DELETE.

The screenshot shows the Microsoft Azure Kubernetes Services Overview page for a secret named 'repo-1295282847'. The URL field contains a long Base64 encoded string: 'aHR0cHM6Ly9naXRodWIuY29tL1Zpdmllbjg3L215d2VzGvtby5naXQ='.

```
PS E:\projects\terraformProj\nodeWebApp\Rolling> argo cd repo list
TYPE NAME REPO INSECURE OCI LFS CREDs STATUS MESSAGE PROJECT
git https://github.com/Vivien87/symwebdemo.git false false false false Successful
https://github.com/Vivien87/symwebdemo.git [System.Text.Encoding]:UTF8.GetString([System.Convert]::FromBase64String("aHR0cHM6Ly9naXRodWIuY29tL1Zpdmllbjg3L215d2VzGvtby5naXQ="))
PS E:\projects\terraformProj\nodeWebApp\Rolling>
```